

---

**Nama : Muhammad Alfarezzi Fallevi**

**Kelas : 4IA17**

**NPM : 50421905**

## **Actifity 2 Minggu Ke-2**

### **Rekayasa Perangkat Lunak 2**

---

**1. Jelaskan anatomi dasar dari class dalam java. Apa saja komponen utama yang ada dalam class dan bagaimana fungsinya? Berikan contoh sederhana dari sebuah class dengan atribut dan method didalamnya.**

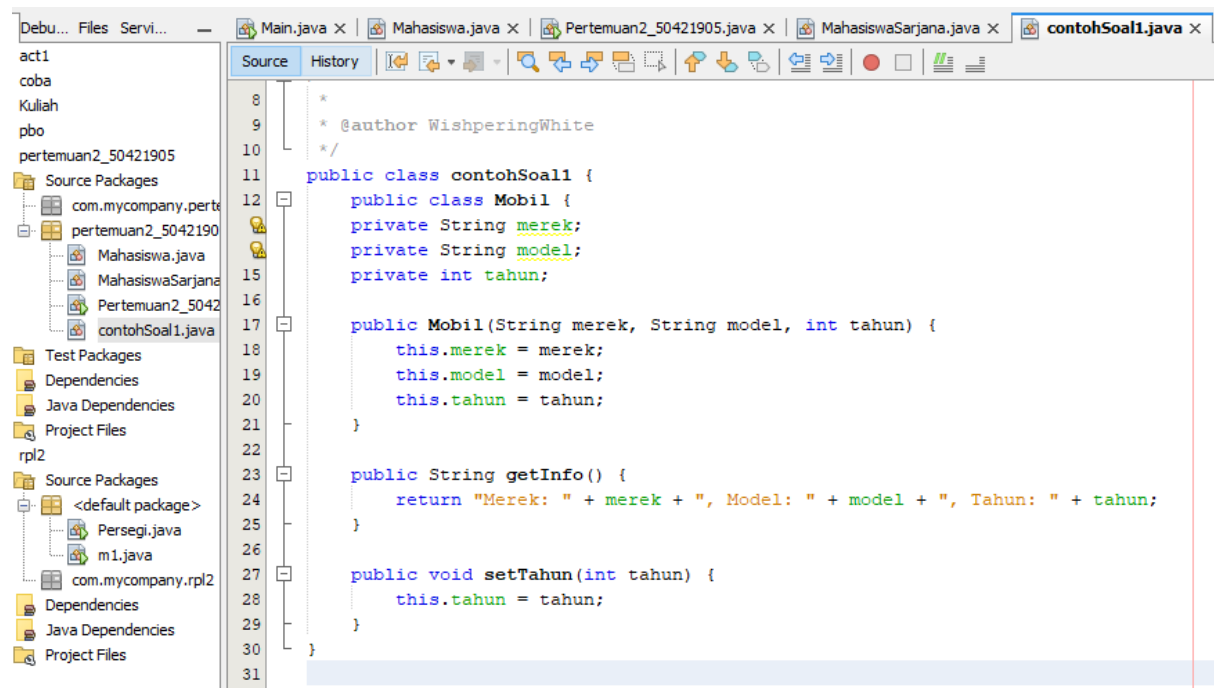
**2. Screenshot kode, output, dan beri penjelasan singkat dari program yang telah dibuat pada video**

---

**1. Sebuah class dalam Java memiliki beberapa komponen utama:**

1. Deklarasi Class
  - Menggunakan keyword `class` diikuti nama class
  - Biasanya menggunakan PascalCase (huruf pertama kapital)
2. Atribut/Fields/Properties
  - Variabel yang menyimpan data/state dari objek
  - Biasanya dideklarasikan di awal class
  - Dapat memiliki access modifier (`public`, `private`, `protected`)
3. Constructor
  - Method khusus untuk inisialisasi objek
  - Namanya sama dengan nama class
  - Dipanggil saat objek dibuat dengan keyword `new`
4. Method
  - Fungsi yang merepresentasikan behavior/perilaku dari class
  - Dapat memiliki parameter dan return value
  - Dapat diakses melalui objek

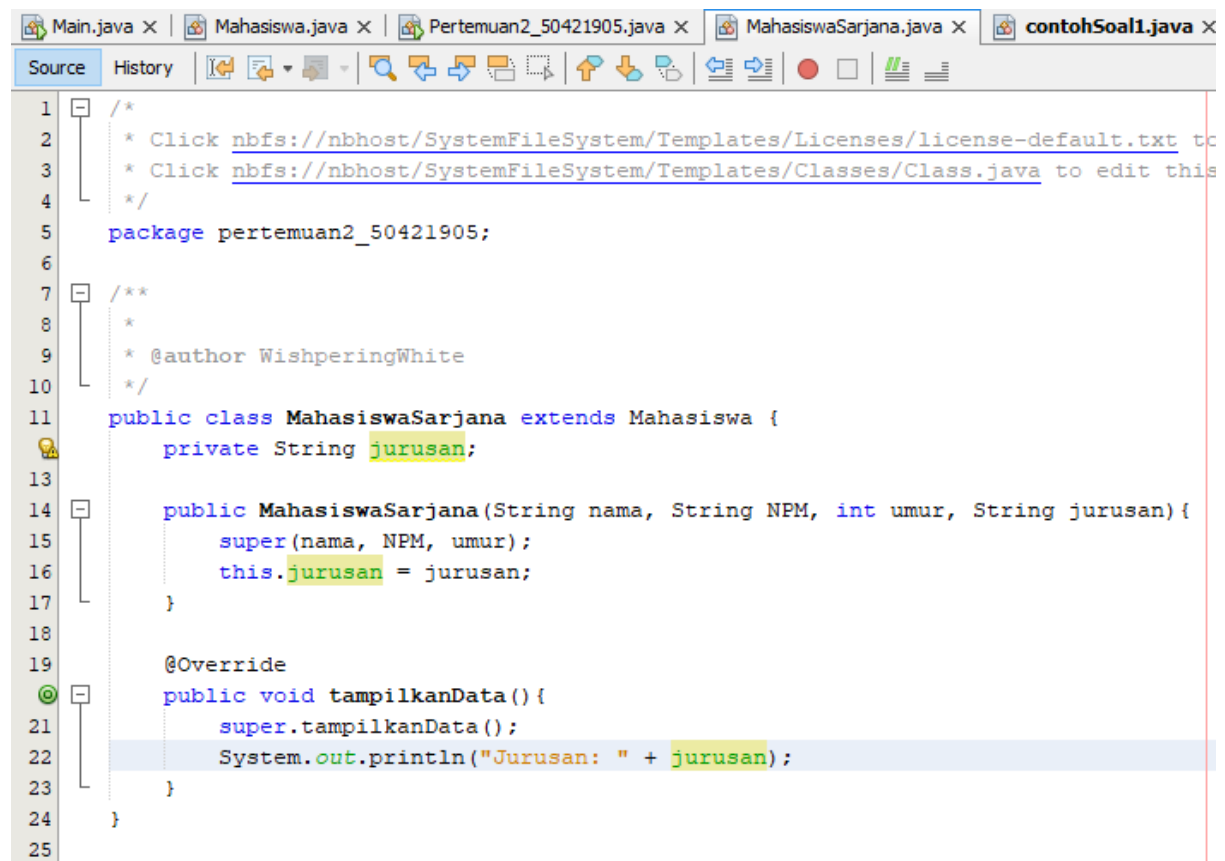
## Contoh



The screenshot shows an IDE with the 'contohSoal1.java' file open. The code defines a class 'contohSoal1' containing a nested class 'Mobil'. The 'Mobil' class has private attributes 'merek', 'model', and 'tahun', a constructor, a 'getInfo()' method, and a 'setTahun()' method. The IDE's left sidebar shows a project structure with 'Source Packages' and 'Test Packages'.

```
8      *
9      * @author WishperingWhite
10     */
11     public class contohSoal1 {
12     public class Mobil {
13     private String merek;
14     private String model;
15     private int tahun;
16
17     public Mobil(String merek, String model, int tahun) {
18     this.merek = merek;
19     this.model = model;
20     this.tahun = tahun;
21     }
22
23     public String getInfo() {
24     return "Merek: " + merek + ", Model: " + model + ", Tahun: " + tahun;
25     }
26
27     public void setTahun(int tahun) {
28     this.tahun = tahun;
29     }
30     }
31 }
```

## 2. Code



The screenshot shows an IDE with the 'MahasiswaSarjana.java' file open. The code defines a class 'MahasiswaSarjana' that extends 'Mahasiswa'. It has a private attribute 'jurusan', a constructor, and an overridden 'tampilkanData()' method. The IDE's left sidebar shows a project structure with 'Source Packages' and 'Test Packages'.

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4  */
5  package pertemuan2_50421905;
6
7  /**
8   *
9   * @author WishperingWhite
10  */
11  public class MahasiswaSarjana extends Mahasiswa {
12  private String jurusan;
13
14  public MahasiswaSarjana(String nama, String NPM, int umur, String jurusan) {
15  super(nama, NPM, umur);
16  this.jurusan = jurusan;
17  }
18
19  @Override
20  public void tampilkanData() {
21  super.tampilkanData();
22  System.out.println("Jurusan: " + jurusan);
23  }
24  }
25  }
```

- Inheritance: Menggunakan extends untuk mewarisi properti dan method dari class Mahasiswa
- Super keyword:
  - super(nama, NPM, umur) memanggil constructor parent
  - super.tampilkanData() memanggil method parent
- Method Overriding: menggunakan annotation @Override untuk menimpa method tampilkanData() dari parent

```

7  /**
8   *
9   * @author WishperingWhite
10  */
11  public class Pertemuan2_50421905 {
12
13      public static void main(String[] args){
14          Mahasiswa mahasiswa = new Mahasiswa("Levi", "50421905", 22);
15          mahasiswa.tampilkanData();
16          System.out.println();
17
18          MahasiswaSarjana mahasiswaSarjana = new MahasiswaSarjana("Firman", "50421212", 22, "Informatika");
19          mahasiswaSarjana.tampilkanData();
20          System.out.println();
21      }
22  }
23

```

- Inheritance (Pewarisan), dimana MahasiswaSarjana kemungkinan merupakan turunan dari class Mahasiswa
- Pembuatan dua objek berbeda:
  - Objek mahasiswa dengan data: nama "Levi", NIM "50421905", umur 22
  - Objek mahasiswaSarjana dengan data tambahan berupa program studi "Informatika"
- Kedua objek memanggil method tampilkanData() untuk menampilkan informasi masing-masing

```

0  L  */
1  public class Mahasiswa {
2      private String nama;
3      private String NPM;
4      private int umur;
5
6      public Mahasiswa(String nama, String NPM, int umur) {
7          this.nama = nama;
8          this.NPM = NPM;
9          this.umur = umur;
10     }
11
12     public void tampilkanData() {
13         System.out.println("Nama: " + nama);
14         System.out.println("NPM: " + NPM);
15         System.out.println("Umur: " + umur);
16     }
17 }

```

- Mendefinisikan 3 variabel private: nama (String), NPM (String), dan umur (int)
- Private artinya hanya bisa diakses di dalam class ini saja
- Method khusus untuk membuat objek Mahasiswa baru
- Menerima 3 parameter: nama, NPM, dan umur
- this merujuk ke atribut class, untuk membedakan dengan parameter
- Method untuk menampilkan data mahasiswa ke layer

## Output

Output - Run (pertemuan2\_50421905)

```

--- exec-maven-plugin:3.0.0:ex
Nama: Levi
NPM: 50421905
Umur: 22

Nama: Firman
NPM: 50421212
Umur: 22
Jurusan: Informatika

```